LiveView ActiveX Control Properties

BSTR IPAddress;

IP address of the live streaming server

BSTR UserName;

User name to login the live streaming server

BSTR Password;

Password for the user name

long Port;

TCP port of the live streaming server

LiveView ActiveX Control Methods

BOOL Connect();

Connect to remote server

Return Values:

TRUE	Successfully to connect to server
FALSE	Failed to connect to server

BOOL Disconnect();

Disconnect from server

Return Values:

TRUE	Successfully disconnect from server
FALSE	Failed to disconnect from server

BOOL ConnectToCamera(long iView, long iCameraIndex);

Connect to specified camera

Parameters:

iView	which view to display
iCameraIndex	which camera to connect

Return Values:

TRUE	Successfully connect to camera
FALSE	Failed to connect to camera

BOOL SetActiveView(long iView);

Focus on specified view, need to call ConnectToCamera() first.

Parameters:

iView	Focus on which view

Return Values:

TRUE	Successfully switch the active view
FALSE	Failed to switch the active view and possibly due to not connected to camera

long GetActiveView();

Get the focused view

Return Values:

Active view index

long GetCameraIndexFromView(long iView);

Get camera's index from view

Parameters:

iView	Which view is selected.

Return Values:

Camera index on the specific view

If not connected, return -1

long GetCameraCount();

Get camera number

Return Values:

Camera total number

BSTR GetCameraName(long iCameraIndex);

Get the camera name

Parameters:

iCameraIndex	Camera's Index

Return Values:

Camera Name

BSTR GetCameraModel(long iCameraIndex);

Get the camera model

Parameters:

iCameraIndex	Camera's Index

Return Values:

Camera Model

void EnableAudio(long iView, BOOL bEnable);

Enable audio of camera

Parameters:

iView	Which view is selected
bEnalbe	TRUE: audio enabled
	FALSE: audio disabled

void ChangeResolution(long iWidth, long iHeight);

Change the video window size.

Parameters:

iWidth	video window width
iHeight	video window height

void SetOSDConfig(long iCmd, LPCTSTR fontStyle, int iFontSize, int iColorR, int iColorG, int iColorB, BOOL bHasEdge, BOOL bBold, int iBColorR, int iBColorG, int iBColorB, iBKTransparency);

Set Camera's OSD Config.

Parameters:

iCmd	0x00: Disabled
	0x01: Set Date
	0x02: Set Time
	0x08: Set Camera Name
	0x10: Set Bitrate.
fontStyle	OSD font style
iFontSize	OSD font size
iColorR	OSD font color R (0-255)
iColorG	OSD font color G (0-255)
iColorB	OSD font color B (0-255)
bHasEdge	0: Has edge
	1: Doesn't has edge
bBold	0: Normal
	1: Bold
iBColorR	OSD background color R (0-255)
iBColorG	OSD background color G (0-255)
iBColorB	OSD background color B (0-255)
iBkTransparency	OSD background transparency (0-255)

long GetRawImageWidth(long iView);

Get camera’s raw image’s width

Parameters:

iView	Get which view’s raw image

Return Values:

Raw image’s width

long GetRawImageHeight(long iView);

Get camera’s raw image’s height

Parameters:

iView	Get which view’s raw image

Return Values:

Raw image’s height

VARIANT GetRawImage(long iView, long iVideoFormat);

Get camera’s raw image

Parameters:

iView	Get which camera’s raw image
iVideoFormat	0: YUV12
	1: RGB

Return Values:

VARIANT.parray contains the whole raw image

long GetDICount();

Get digital input count

Return Values:

Total digital input number

long GetDOCount();

Get digital output count

Return Values:

Total digital output number

BSTR GetDIName(long iInputIndex);

Get digital input name.

Parameters:

iInputIndex	which input is selected.

Return Value:

Digital input name.

BSTR GetDOName(long iOutputIndex);

Get digital output name.

Parameters:

iOuputIndex	which output is selected.

Return Value:

Digital output name

Long GetDIStatus(long iInputIndex);

Check digital input status.

Parameters:

iInputIndex	which input is selected.

Return Value:

0:	Low
1:	High

Long GetDOStatus(long iOuputIndex);

Check digital output status.

Parameters:

iOuputIndex	which output is selected.

Return Value:

0:	Low
1:	High

void SetDOStatus(int iOutputIndex, long iValue);

Set digital output status.

Parameters:

iOutputIndex	Which output needs to set enabled.
iValue	0: Low 1: High

long GetPresetCount(long iView);

Get preset count

Parameters:

iView	which camera's total preset

Return Values:

Preset count

BSTR GetPresetName(long iView, long iPresetNum);

Get preset’s name

Parameters:

iView	View index
iPresetNum	Preset’s number

Return Values:

Preset name

void PTZPresetGO(long iView, long iPresetNum);

Go to preset point

Parameters:

iView	View index
iPresetNum	Preset’s number

BOOL SetPreset(long iView, long iPresetNum, LPCSTR PresetName);

Add a camera preset on specific view

Parameters:

iView	View index
iPresetNum	Preset number(-1 means delete all preset)
PresetName	Specified preset name

Return Values:

TRUE	Successfully to set preset
FALSE	Failed to set preset

void PTZZoomTele(BOOL bStart);

Zoom in the camera, need to call SetActive() first.

Parameters:

bStart	0: stop to zoom in
	1: star to zoom in

void PTZZoomWide(BOOL bStart);

Zoom out the camera, need to call SetActive() first.

Parameters:

bStart	0: stop to zoom out
	1: start to zoom out

void PTZUp(BOOL bStart);

Move the camera up, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera up

1: start o move the camera up

void PTZDown(BOOL bStart);

Move the camera down, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera down

1: start to move the camera down

void PTZLeft(BOOL bStart);

Move the camera left, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera left

1: start to move the camera left

void PTZRight(BOOL bStart);

Move the camera right, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera right

1: start move the camera right

void PTZUpLeft(BOOL bStart);

Move the camera up and left, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera up and left

1: start to move the camera up and left

void PTZUpRight(BOOL bStart);

Move the camera up and right, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera up and right

1: start move the camera up and right

void PTZDownLeft(BOOL bStart);

Move the camera down and left, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera down and left

1: start to move the camera down and left

void PTZDownRight(BOOL bStart);

Move the camera up and left, need to call SetActive() first.

Parameters:

bStart 0: stop to move the camera down and right

1: start to move the camera down and right

void PTZHome();

Move the camera to home position, need to call SetActive() first.

void PTZPatrol(BOOL bStart);

Set camera to Patrol through default preset, need to call SetActive() first.

Parameters:

bStart 0: Start patrol

1: Stop patrol

void PTZFocusNear(BOOL bStart);

Set the camera to focus near, need to call SetActive() first.

Parameters:

bStart 0: Start to focus near

1: Stop to focus near

void PTZFocusFar(BOOL bStart);

Set the camera to focus far, need to call SetActive() first.

Parameters:

bStart 0: Start to focus far

1: Stop to focus far

void Play();

Play live video from remote server on active view

void Stop();

Stop the live video from remote server on active view

void Drop();

Drop the live video from remote server on active view

BOOL Snapshot(long iView, LPCTSTR FileName);

Save the live video snapshot, need to call ConnectToCamera() first.

Parameter:

iView:	Which Camera’s snapshot
FileType:	.bmp
	.BMP
	.jpg
	.JPG

Return Values:

TRUE	Successfully to save a snapshot
FALSE	Failed to save a snapshot

void SetView1x1();

Set 1x1 window

void SetView2x2();

Set 2x2 window

void SetView3x3();

Set 3x3 window

void SetView4x4();

Set 4x4 window

void SetViewFullScreen();

Set full screen

BOOL DuplicateCamera(long iCameraIndex);

Duplicate the specified camera

Parameters:

iCameraIndex	which camera to connect

Return Values:

TRUE	Successfully duplicate the camera
FALSE	Failed to duplicate the camera

BOOL RemoveDuplicateCamera(long iCameraIndex);

Remove the duplicated camera

Parameters:

iCameraIndex	which duplicated camera to be remove

Return Values:

TRUE	Successfully remove the camera
FALSE	Failed to remove the camera

BOOL GetPTZCapability(long iCameraIndex, long iParam);

Get the supported status of the specified PTZ capability of the camera

Parameters:

iCameraIndex	which camera to get the information
iParam	0:Enable
	1:Buildin
	2:Pane
	3:Tilt
	4:Zoom
	5:Focus
	6:Preset
	7:Lilin
	8:AutoPan
	9:AreaZoom
	10:SpeedDomeOSDMenu

Return Values:

TRUE	Supported
FALSE	Not supported

BOOL GetCameraCapability(long iCameraIndex, long iParam);

Get the supported status of the specified capability of the camera

Parameters:

iCameraIndex	which camera to get the information
iParam	0:Audio
	1:Talk
	2:PTZ
	3:DIO

Return Values:

TRUE	Supported
FALSE	Not supported

void EnableTalk(long iCameraIndex);

Start to send input audio to camera

Parameters:

iCameraIndex	which camera to send audio

void DisableTalk();

Stop to send input audio to camera

LiveView ActiveX Control Events

void OnEvent(long iEventID, long iParameter);

System event handler

Parameter:

iEventID:	0: OnConnectSuccess
	1: OnConnectFailed
	2: OnDisconnect
	3: OnCameraConencted
	4: OnCameraStopped
	5: OnCameraDropped
	6: Reserved
	7.OnTalkDisconnected
	8.OnTalkConnected
iParameter:	Cameraindex if the EventID is 4, 5, 6

void OnServerEvent(BSTR DateTime, int iDeviceIndex, BSTR EventType, BSTR auxiliaryCode, BSTR description, BSTR sourceName);

System event handler

Parameter:

DateTime:	Event occurring time
iDeviceIndex	The device trigger this event
EventType	Event type
auxiliaryCode	Reserved
description	the description of event
sourceName	Reserved